Intelligent Automation
& Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.037694
Article

A Secure Microgrid Data Storage Strategy with Directed Acyclic Graph Consensus Mechanism

Jian Shang[1,2,*], Runmin Guan[2] and Wei Wang[2]

[1]College of Computer and Information, Hohai University, Nanjing, 211100, China
[2]Division of Research and Innovation, Jiayuan Technology Co., Ltd., Nanjing, 211100, China
*Corresponding Author: Jian Shang. Email: shangjian@jiayuantech.com

Received: 14 November 2022; Accepted: 19 May 2023; Published: 11 September 2023

Abstract: The wide application of intelligent terminals in microgrids has fueled the surge of data amount in recent years. In real-world scenarios, microgrids must store large amounts of data efficiently while also being able to withstand malicious cyberattacks. To meet the high hardware resource requirements, address the vulnerability to network attacks and poor reliability in the traditional centralized data storage schemes, this paper proposes a secure storage management method for microgrid data that considers node trust and directed acyclic graph (DAG) consensus mechanism. Firstly, the microgrid data storage model is designed based on the edge computing technology. The blockchain, deployed on the edge computing server and combined with cloud storage, ensures reliable data storage in the microgrid. Secondly, a blockchain consensus algorithm based on directed acyclic graph data structure is then proposed to effectively improve the data storage timeliness and avoid disadvantages in traditional blockchain topology such as long chain construction time and low consensus efficiency. Finally, considering the tolerance differences among the candidate chain-building nodes to network attacks, a hash value update mechanism of blockchain header with node trust identification to ensure data storage security is proposed. Experimental results from the microgrid data storage platform show that the proposed method can achieve a private key update time of less than 5 milliseconds. When the number of blockchain nodes is less than 25, the blockchain construction takes no more than 80 mins, and the data throughput is close to 300 kbps. Compared with the traditional chain-topology-based consensus methods that do not consider node trust, the proposed method has higher efficiency in data storage and better resistance to network attacks.

Keywords: Microgrid; data security storage; node trust degree; directed acyclic graph data structure; consensus mechanism; secure multi-party computing; blockchain

1 Introduction

Microgrid is a vital power supply solution for offshore islands, industrial parks, and remote mountainous areas. With the widespread application of Internet of Things technology in microgrids, the data amount in microgrids has shown an explosive growth trend [1]. However, there exist a potential risk of data leakage with traditional centralized management schemes for data exchange during microgrid operations. This risk usually arises due to the third-party platforms may copy, store, or resell the data. Thus, there is an urgent need for a secure data storage management method [2,3].

The centralized data management methods are insufficient in ensuring adequate data security and privacy protection for microgrids with massive and decentralized terminals [4,5]. In addition, since the microgrids consist of a large number of remote terminals with relatively weak computing and communicating capabilities, the traditional centralized data management schemes which rely on high computation and communication capabilities can hardly guarantee data communication and storage security [6]. Additionally, given that most applications in microgrids are highly sensitive to delays, specific requirements must be met regarding the storage delays of microgrid data [7].

Currently, blockchain based secure data storage schemes have been widely discussed in the majority of existing literature considering the outstanding privacy protection advantages due to immutability, data consistency, and characteristics such as decentralized trust and cross-domain identity authentication [8]. Besides, the geographically distributed nature is in line with the decentralized data management characteristics of the blockchain, which means that there is an inherent advantage in applying blockchain technology to the secure microgrid data management. However, it should be noted that in traditional chain-topology-based blockchains, a large number of nodes can consume significant communication bandwidth and network resources, resulting in low state update efficiency for the entire chain when new nodes are added. Moreover, as a typical cyber-physical system, if storage nodes involved in building the blockchain are attacked by hackers, there exists a potential data leakage risk. Unfortunately, most existing literature on blockchain construction algorithms have overlooked the trustworthiness of nodes. If sensitive data related to microgrid operations are obtained by attackers, it could result in severe accidents such as equipment misoperations or even blackouts.

To this end, this paper proposes a secure storage management method for microgrid data considering node trust and utilizing directed acyclic graph consensus mechanism (DAG). Unlike existing consensus algorithms with a chained topology, the proposed DAG-based consensus mechanism has higher consensus efficiency due to parallelism advantages. Additionally, to mitigate the data leakage risk posed by compromised nodes in blockchain construction, a secure multi-party computing-based Hash updating algorithm is developed, which takes into account the trustworthiness of candidate nodes. Finally, case study results show that the proposed method with DAG topology can effectively enhance the security and efficiency of microgrid data storage.

2 Related Work

Ensuring the safe and stable operations of power systems is crucial for promoting economic development, preserving public safety, and maintaining political stability. Given its paramount importance, the secure storage and trusted access of power system operation data has garnered considerable attention [9]. Currently, most microgrid data storage schemes utilize a centralized framework. Based on the data irrelevant redaction to services and bounded artificial noise addition, Reference [10] proposed a security storage and access framework for the electrical consumption data. Reference [11] developed an anonymous and secure two-factor authentication and key agreement scheme to deal with the remote data storage and access in cloud computing architecture. However, this method heavily relies on the reliabilities of access terminals, which may impede its effectiveness in maintaining data security. Reference [12] proposed a p-sensitive k-anonymity privacy protection attribute for the personal information privacy protection to prevent information leakage from the perspective of model gains. However, the private key generation process of identity-based key system is relatively complicated, and the execution process is relatively cumbersome.

The decentralization feature of blockchain technology has facilitated its rapid development and extensive application. Significant efforts have been devoted to applying the blockchain technology to data privacy protection in the power systems [13]. Reference [14] proposed an integrated and lightweight blockchain model that established an overlay network, integrated resources into a public blockchain, and conducted dedicated security and privacy protection verification. Although this model had high security requirements for public blockchains, the integration efficiency on complex resource type is relatively low. Reference [15] proposed a blockchain-based big data privacy protection model for user behavior data to improve the performance and efficiency of traditional privacy protection schemes. However, the proposed scheme did not consider the impact of network loads on the response speed of blockchain. Reference [16] proposed an edge blockchain-assisted lightweight privacy protection scheme for smart grids that combined edge computing and blockchain, generating better resistance to attacks and lower computing and communication overhead. However, the multi-node data processing performance remains to be improved. Reference [17] applied the blockchain technology to store the electrical data collected by the wireless network and designed a blockchain-based secure power transaction mechanism for smart grids to ensure communication and storage security. The blockchain technology, however, requires the participation of all nodes in the network during the consensus stage, which consumes massive network resources. In addition, once the candidate storage nodes competing for building the blockchain are attacked by the hackers, the data security of microgrid system will be greatly reduced. In this case, the operation stability of the microgrid system may be affected. For instance, in 2015, the monitoring subsystem in the substation of the Ukrainian power system was deliberately hacked by attackers, leading to a blackout accident [5]. Therefore, evaluating the trust degree of the storage node before participating in the blockchain building can effectively enhance the data security of microgrid. In addition, the blockchains in literature mentioned above adopted the chained topology, which may cause significant time consumption due to consensus algorithms. For time-sensitive applications such as voltage control and frequency regulation in microgrids, the additional delays may degrade control performance and jeopardize the safe operation of the microgrid.

To deal with low consensus efficiency and potential cyber attack risks in current blockchain based microgrid data storage schemes with chained topology, this paper proposes a microgrid data security storage method based on node trust and directed acyclic graph (DAG) consensus mechanisms in an attempt to address the problems such as high multi-node resource consumption and low update efficiency of new joining nodes in the current blockchain-based data security storage methods. The contributions are summarized as follows:

(1) An algorithm using a directed acyclic graph (DAG) structure is developed to enable the confirmation of a large number of transactions within a short period, which is not feasible with the traditional chained blockchain topology. The proposed DAG method allows for simultaneous transaction confirmation by multiple users and significantly improves the timeliness of data storage.

(2) A blockchain header hash value update mechanism taking into consideration the trustworthiness of nodes in the chain-building process is proposed. This mechanism enhances the ability of nodes to resist cyber attacks, and ensures the security of data storage by generating the hash value of the

blockchain header based on the trustworthiness of the participating nodes. This approach significantly improves the nodes' candidate's ability to resist cyber attacks during the blockchain construction process.

3 Proposed Method

3.1 Overall Framework

The ongoing construction and development of Power Internet of Things have resulted in a widespread distribution of various data sources within the microgrid, which poses a challenge to centralized management. To address this issue, as shown in Fig. 1, edge computing servers are deployed at the network edge to process nearby data, along with the deployment of blockchain technology to prevent security issues caused by excessive data dispersion. Furthermore, the combination of blockchain and cloud storage is employed to store the data index table in the blockchain, while the corresponding data is stored in the cloud database [18]. This storage method non only can release a large amount of storage space in the blockchain, but also enhances the accessing and sharing efficiency.

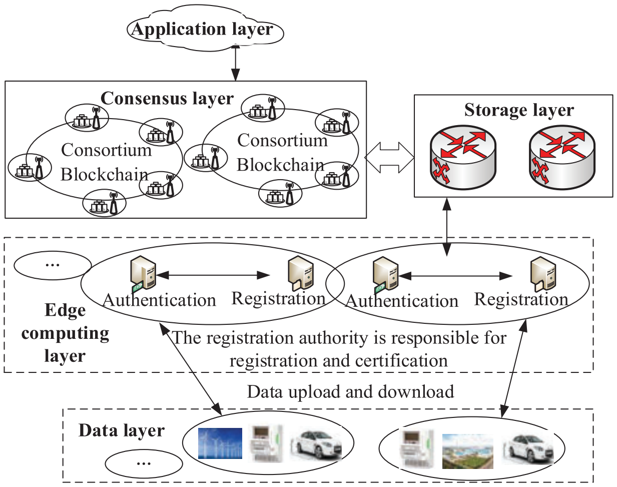

Figure 1: Edge computing-based microgrid data storage system architecture

The data layer is mainly responsible for collecting, cleaning, encrypting, and packaging the microgrid data, while the edge computing layer is charge of user authentication and registration. The consensus and storage layers are the core components of the proposed method, while the application layer mainly initiates data access applications. Once the users' identities are verified, the smart contract will be automatically triggered to enable the data sharing [19]. Each layer plays a vital role and works collaboratively to form a microgrid data security storage platform through mutual collaboration. The structure of the secure storage process is shown in Fig. 2.

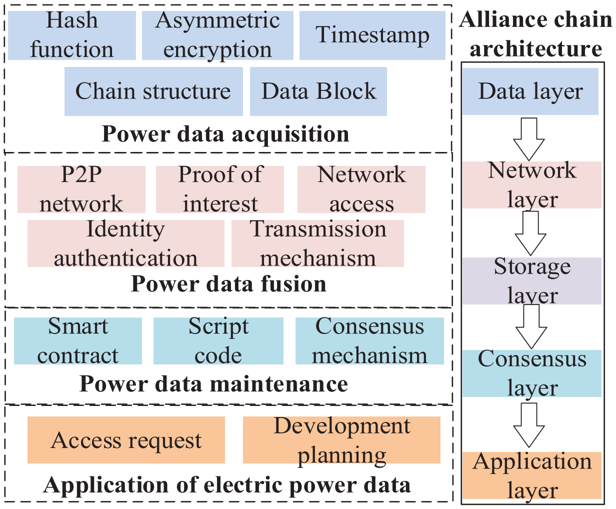

Figure 2: Structure division of security storage process for microgrid data

Each storage node in the microgrid is a candidate bookkeeper that stores the data generated during microgrid operation. Since the storage nodes may vary in resistance to cyber-attacks, it is necessary to evaluate in advance the trustworthiness of storage nodes in the storage layer. A DAG-based consensus algorithm is used at the consensus layer to realize the secure storage and management of microgrid data and improve the timeliness of data storage simultaneously.

(1) Data layer: As a physical area, the data layer is a fundamental module on the microgrid data security storage platform. It is designed to collect data information from the microgrid equipment, such as operation status, energy type, source address, receiving time. Then the collected data are transmitted to the edge computing layer for processing.

(2) Edge computing layer: This layer is responsible for extracting the features of different types of data uploaded from the data layer. At this layer, the features will be described with a fixed-length mathematical base using some specific technologies such as hash function, asymmetric encryption, Markle root value [20,21]. After unifying the formats of microgrid data, the data information and transaction codes are encapsulated into blocks with time stamps. These new blocks will be linked to the main block with the longest chain to form a new block node. The edge computing layer also includes blockchain technology elements such as networking mode and data authentication protocol, which allow each block node to participate in the transmission and verification of microgrid data information. That is to say, each node undertakes not only the network routing, but also the block information authentication and the data transmission.

(3) Storage layer: This layer stores data and returns the storage index to the consensus layer. With each node reaching a consensus, the data will be stored in this layer. Then, the storage addresses will be returned to consortium chain storage in the consensus layer.

(4) Consensus layer. Similar to the blockchain algorithms discussed in current literature (e.g., [22,23]), a consensus algorithm requiring more than 50% of the nodes to confirm transaction security is designed to guarantee the data storage security. Considering that there usually exists a high time consumption in the consensus process with traditional chained-topology, a directed acyclic graph (DAG) consensus mechanism adopting a parallel working mode is designed in this paper. It should

also be pointed out that the proposed consensus mechanism fully takes the trust degree of the storage node into consideration to ensure the data security of the microgrid system.

(5) Application layer: Various applications depending on the microgrid data stored in the alliance chain-based secure storage platform are encapsulated at this layer. This layer also serves an interactive processing platform for the operating data in the microgrid, helping realize the efficient and secure information exchanges among different enterprise departments for the microgrid enterprises, which thereby, ensures the normal microgrid operations [24,25].

3.2 DAG-Based Blockchain Consensus Algorithm

In the traditional storage structure with chained topology, there is only one single chain, which makes it impossible for the blockchain to conduct the consensus algorithm in parallel. Moreover, the computing process in data storage is quite time-consuming for the chained-topology blockchain structure. Therefore, it is not suitable for frequent data requests in real-time microgrid applications. To address these issues, we propose a consensus algorithm using a directed acyclic graph (DAG) topology to replace the traditional chained topology, attempting to overcome the problems of low efficiency and huge resource occupancy in the traditional blockchain consensus algorithms. The storage structure with DAG topology has an apparent advantage, namely the consensus algorithm can be conducted parallelly at the same time, which helps improve the data storage efficiency effectively. In short, the proposed DAG-based consensus algorithm can overcome the weakness of the traditional chain structure blockchain, which is its inability to confirm a large number of transactions in a short time, and the proposed algorithm can help multiple users to confirm transactions simultaneously. The DAG-based blockchain node structure is shown in Fig. 3.

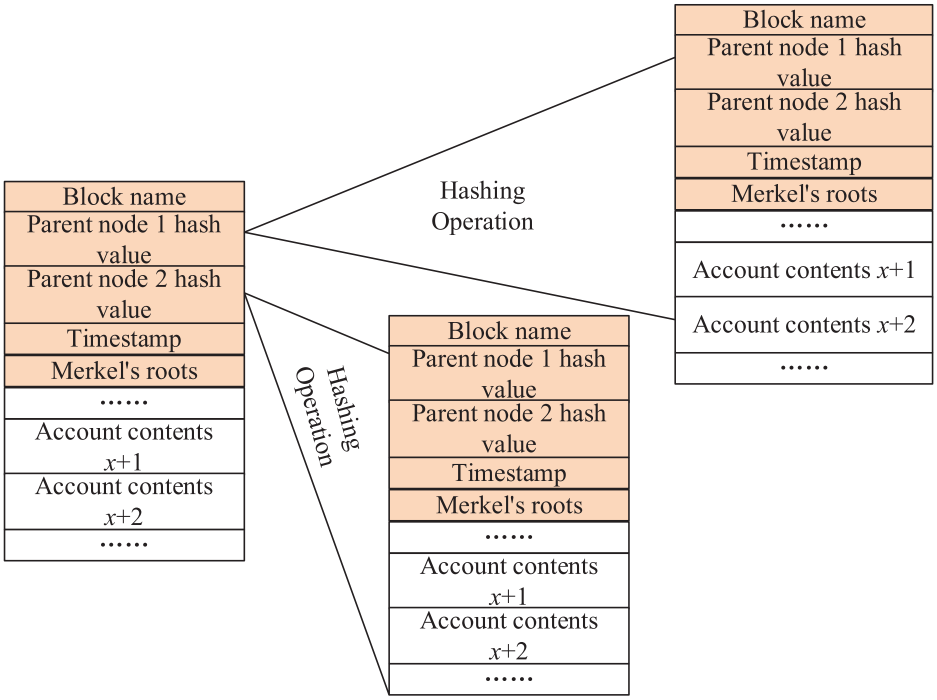

Figure 3: Block chain structure based on DAG structure

According to the design of Tangle protocol, each block contains one piece of transaction information. When a new transaction is generated, the corresponding block randomly selects two historical transactions as its parent transaction. After verifying the legality of the parent transaction, the new block's own transaction information will be written into the block and connected to its parent transaction blocks. In our proposed consensus algorithm, the hash value in the block considers the trustworthiness of the node. Moreover, the account contents usually contain the data characteristic fields such as the type, address, timestamp of the stored data [26,27].

The proposed consensus algorithm consists of three stages. In the first stage, both parties in the transaction generate a transaction order and complete the creation of initial transaction block. The transaction contents of the transaction block are signed to ensure the legality of the transaction block. The second is the block verifying and signing stage of miners. The completion of signature means the completion confirmation for legality of transaction block contents. In the third stage, the last miner participated in the signature broadcasts the transaction block to the entire blockchain network. Then all nodes synchronize the new block after receiving the information sent by the last miner. Finally, the transaction is successfully recorded in the blockchain.

Specifically, in the first stage, both parties in the transaction need to negotiate and complete signing the transaction contents. After the signing is completed, the transaction initiator will create the blocks and broadcasts it to the blockchain network. The created block is then verified for legality and waits to be signed by the miners. The generated block needs to be attached to the chain of ID block where the transaction initiator is located.

In the second stage, the miners need to complete the signature verification of the initial transaction block generated in the first stage. The main steps are as follows:

(1) The miner firstly checks the rst field. If the value of this field is 0, the block will be rejected and broadcasted to other miners in the network. The transaction block generation algorithm of this block on the miner side will be terminated. The mathematical description is:

$$T_{\mathrm{rst}}(b) = \begin{cases} \varphi(T - \mathrm{Block}), & \mathrm{rst} = 0 \\ T_{\mathrm{p-list}}(b), & \mathrm{rst} \neq 0 \end{cases} \tag{1}$$

where b is the block information; T-Block is the block; $\varphi()$ is the broadcast block function; $T_{\text{p-list}}$ is the second step of block operation; p-list is the miner signature list of previous block.

(2) Check the miner's signature list of the previous block. If the miner's ID has appeared on this list, it means that the miner has participated in signing the blocks. In this scenario, this block should be rejected and broadcasted to other miners in the network. The transaction block generation algorithm of this block on the miner's side will be terminated.

(3) Determine whether the signed miner ID(p) has appeared on the previous miner's signature list. The function $h()$ indicates that the miner has violated the operating rules if a match is found between the signed miner ID and the miner ID on the previous miner's signature list. The block is rejected and broadcasted to other miners in the network, and the transaction block generation algorithm for this block on the miner's side is terminated. The mathematical description is:

$$T_{\mathrm{check}} = h(\mathrm{m} - \mathrm{list}, \mathrm{p} - \mathrm{list}, p) \tag{2}$$

where m-list is the signature list of the miners in this block, p is the order of the miner.

(4) The rst value is decremented by 1.

(5) Verify whether the signatures of both parties are legal through the public keys of both parties in the block. If the verification of the validity of signature fails, the miner will reject the signature and broadcast the block to other miners in the network. The transaction block generation algorithm of this block on the miner's side will be terminated. The mathematical expression is:

$$T_{\text{signature}} = \kappa\left(C\left(T - \text{Block}, g_a, \text{Pkey}_a\right), C\left(T - \text{Block}, g_b, \text{Pkey}_b\right)\right) \tag{3}$$

where κ represents the verification function of the verification results of both parties' signatures; $C()$ represents the verification of the legitimacy for the miner's signature. g_a and g_b are the digital signatures of the two parties in the transaction; Pkey_a and Pkey_b are the public keys of the two parties in the transaction.

(6) Check whether the block is placed in the correct position. If the check fails, then reject the block and broadcast the block to other miners in the network. The transaction block generation algorithm of this block on the miner's side will be terminated.

(7) Check whether the signatures of other miners in the miner's signature list are legal. If any signature verification fails, i.e., if the block signature fails, the block will be rejected and broadcasted to other miners in the network. The transaction block generation algorithm of this block on the miner's side will be terminated.

(8) The miner signs the block and uses elliptic curve digital signature (ECDS) method to generate a digital signature (b, g) through block b, random number r and private key $\text{key}_{\text{private}}$. The mathematical expression is:

$$T_{\text{sign}} = \text{ECDS}\left(b, r, \text{key}_{\text{private}}\right) = (b, g) \tag{4}$$

(9) The miner needs to confirm whether it is the last one to participate in the signing. In other words, the miner should check the rst field (i.e., the number of signed miners). If rst $\neq 0$, the miner is not the last one to participate in the signing, and the block will be broadcasted to other miners in the network. Otherwise, the miner's work on this block will be completed. The mathematical expression is:

$$T_{\text{sn}}(\text{rst}) = \begin{cases} \varphi\left(T - \text{Block}\right), & \text{rst} \neq 0 \\ T_{\text{failure}}(\text{b}), & \text{rst} = 0 \end{cases} \tag{5}$$

(10) If the miner is the last one to participate in the signing, it is necessary to check whether the number of signature failures meets the design requirements (no more than 2 times). If the number of signature failures exceeds 2 times, the block signature is declared as a failure. In this case, the transaction initiator should be notified to reorganize the blocks. The mathematical expression is:

$$T_{\text{failure}}(q) = \begin{cases} \varphi\left(T - \text{Block}\right), & q \geq 2 \\ T_{\text{reward}}(q), & q < 2 \end{cases}, \quad q = 10 - \sigma(n) \tag{6}$$

where n is the miner ID on the miner's reward list; the function $\sigma()$ is used to count the number; q is the number of failure.

(11) The last miner creates a reward list to indicate the rewards of all participating miners. When the miner has completed his work, the block will be formally generated and added to the blockchain. Each node in the blockchain will record this block.

In the third stage, the signed block is broadcasted to the blockchain network by the last miner who participated in signing. Each miner who contributed to the signing process will be rewarded at this stage. However, if the last miner attempts to cheat during the reward distribution process, each participating miner can broadcast information about their dishonest behavior to the blockchain network. If more than one-third of the miners participating in the block signing have released the cheating information, the rewards of dishonest miners in the block signing process will be invalidated. Moreover, the first two miners who participated in the signing will generate the transaction blocks, and the untrustworthy information of the dishonest miners will be recorded in the blockchain as part of the transaction contents. Since the proposed algorithm employs a DAG structure, recording the miner's untrustworthy information will not negatively impact the subsequent block generation or the processing efficiency of the transaction blocks.

3.3 Trust Degree Identification Algorithm Based on Secure Multi-Party Computing

In the proposed DAG-based blockchain consensus algorithm, the node hash value fully considers the trust degree of nodes to improve the resistance ability of candidate chain-building nodes against the cyber attacks. To achieve this, a trust degree identification algorithm has been designed in this subsection. Specifically, the trustworthiness of nodes participating in the blockchain construction is evaluated to generate the hash value of blockchain header. Thereby the data storage security can be ensured.

In this paper, the trust value of a node participating in constructing the blockchain is determined from four aspects. Let the weighting coefficient ω_i be the trust degree, which takes values from {0, 1, 2, 3}. The higher trust degree means the node has higher trustworthiness. The main process of the hash value updating considering the trust degree of the participating nodes contains the following three parts.

3.3.1 Trusted Center (TC) System Initialization

Choose two large prime numbers e_1 and e_2 where e_1 and e_2 satisfy $e_2\,|(e_1-1)$. Let d be the generator of the finite field Z_{e_1}. Let n denote the number of all signers and ω_i be the weight of each certificate authority (CA) member (denoted as $\mathrm{CA_i}$) where ω_i is a non-negative integer. Let $AD_\omega = \sum_{i=1}^{n} \omega_i$ and $u = \left(u_1, u_2, \cdots, u_{AD_\omega}\right)$ be a monotonically increasing sequence of integers. In addition, let G represent the message needed to be signed, an integer ε be the threshold, and an integer z be the maximum allowable updating times of private key.

Obviously, the value of AD_ω is positively correlated with the node trust. In extreme cases, where all nodes are untrusted, u is an empty set. In this case, the initialization will fail, that is, all the nodes will not participate in the microgrid data storage. Thus, the proposed algorithm includes a TC, an information service entity ISE, an identity verification server, and a group of all signature authorization centers CA = {CA_1, CA_2, …, CA_n} with weighted trust degrees.

3.3.2 TC Distributes Keys to CA Members Based on Weight

(1) TC allocates secret shares to CA.

For the member CA_i, TC randomly selects the secret number X_c^i according to its weight ω_i. The secret number X_c^i satisfies $0 < X_c^i < [e_2/AD_\omega]$, where $c = \sum_{j=1}^{i} \omega_j - h$ and h are integers satisfying $0 \leq h < \omega_i$.

Randomly select an integer A_c^i satisfying $0 < A_c^i < \frac{1}{AD_\omega}\left[\frac{N}{(z+1)\,e_2^2} - 1\right]$, where N is the smallest ε product of u_i. TC stores each member CA_i into the table H corresponding to its corresponding secret number X_c^i and integer A_c^i. After that, TC puts all {CA_i} secret shares $\left(X_c^i, A_c^i\right)$ in the set X

where $X = \left\{(X_1, A_1), (X_2, A_2), \cdots, \left(X_{AD_\omega}, A_{AD_\omega}\right)\right\}$. Obviously, the higher the trust degree, the higher the number of secret shares of the node, and the lower privacy leakage risk caused by the attack nodes.

(2) Calculate the group public key and child private keys of members in CA according to the secret shares.

TC selects (X_k, A_k) in the set X where $0 < k \leq AD_\omega$, and calculates the shadow ϕ^i_{xc} of secret number X_k with respect to X^i_c:

$$\phi^i_{kc} = (X_k + A_k e_2)\,\mathrm{mod} u_c \tag{7}$$

where $u_c \in \left\{u_1, u_2, \cdots, u_{AD_\omega}\right\}$. The shadow ϕ^i_{xc} of secret number X_k with respect to X^i_c is stored in table H. Each shadow ϕ^i_{xc} is corresponding to the member CA$_i$. Then TC reads all shadows from table H, and calculates the secret number X^i_c corresponding to the child private key of the member CA$_i$, i.e., $I_{ic} = \sum_{k=1}^{AD_\omega} \phi^i_{kc}\ \mathrm{mod} u_c$. TC will then send I_{ic} to CA$_i$ by secure channel. In addition, TC will announce the current group private key Y calculated by $Y = \prod_{k=1}^{AD_\omega} d^{X_k} \mathrm{mod} e_1$. In this case, the group private key has been implicitly generated as $X = \sum_{k=1}^{AD_\omega} X_k$.

(3) Update the child private key of each member in CA.

In order to ensure communication security and meet the private key updating demand of each member in CA, the private key of each member in CA is designed to be updated periodically. During the update process, the group public key Y still remains unchanged to be applicable to the old signatures. This is because the old signature exists in the historically issued weight identifications. Thus, the historically issued identifications can pass the cross-domain authentication verifications. Assuming that the child private key I_{ic} of the member CA$_i$ is updated and the update period is T, the update steps are given as follows:

1) TC selects integers $A^{(T)}_k$ and $0 < k \leq AD_\omega$ satisfying $0 < A^i_c < \dfrac{1}{AD_\omega}\left[\dfrac{N}{(z+1)\,e_2^2} - 1\right]$ which to update the secret share (X_k, A_k) in X.

2) TC uses $A^{(T)}_k$ to calculate the update factor $\phi^{(T)}_{kc}$ by $\phi^{(T)}_{kc} = A^{(T)}_k e_2 \mathrm{mod} u_c$.Then the calculated $\phi^{(T)}_{kc}$ corresponding to X_c is stored in table H.

3) TC uses the updated factor $\phi^{(T)}_{kc}$ to calculate the child private key $I^{(T)}_{ic}$ of updated CA$_i$ according to the following equation:

$$I^{(T)}_{ic} = I^{(T-1)}_{ic} + \sum_{k=1}^{AD_\omega} \phi^{(T)}_{kc}\ \mathrm{mod} u_c \tag{8}$$

3.3.3 CA Updates Hash Values with Weighted Identifications

The issuance process of issuance system is based on the PBFT consensus algorithm. The number of error nodes in PBFT is ν and the number of CA is larger than 3ν. The distributed keys with different weights are used in the process of issuance of identification. The issuance process is as follows:

(1) The identity requester stamps its identity (denoted as R) with a timestamp T to obtain the identity authorization request message G (expressed as $G = R\|T$). The message G is sent to the authentication server (AS) after being encrypted.

(2) The AS receives the encrypted message G from R. After verifying the user's identity, AS encrypts the identity authorization request message G and sends to CA organization through secure channel. In this case, AS can be regarded as part of CA organization.

(3) Let CA_{e_1} in the same trust domain as the users be the master node to issue the identities. After decrypting the message G, CA_{e_1} will generate a new block. In addition, CA_{e_1} will also generate a pre-preparation identifier and send it to other nodes (e.g., CA_r). Then CA_r will get into the prepare state. CA randomly selects $K_{ic} \in Z_{e_1}$, calculates and broadcasts $r_{ic} = d^{K_{ic}}$ mode$_1$ to other members in CA.

(4) When CA_{e_1} receives pre-preparation order, it receives the message G of newly generated block. After CA_{e_1} receives r_{ic} from other members, it will randomly select ε among them to calculate the first signature denoted as $r = \sum r_{ic} = \sum d^{K_{ic}}$ mode$_1$. The calculated result r will be saved in the pre-preparation mark. After the identification has confirmation information agreed by 2ν nodes with the message authentication code (MAC) mechanism, CA_{e_1} will enter the commit state.

(5) The member CA_i calculates its own share signature (denoted as s_{ic}) through the child private key I_{ic} by the following equation:

$$s_{ic} = r \cdot K_{ic} \cdot G + V_{ic} \mathrm{mod} D \tag{9}$$

where $D = \prod_{i=1}^{t} u_i$, $V_{ic} = \frac{D}{u_c} \mu_{ic} I_{ic}$ mod$D$, $\mu_{ic} = \left(\frac{D}{u_c}\right)^{-1}$ modu_c. The calculated result s_{ic} is stamped with a timestamp T to generate the message $G_1 = G \| T \| s_{ic}$ which will be stored in the prepare identifier to obtain the commit identifier. The commit identifier will then be sent to other nodes.

(6) CA_r verifies the consistency of the hash value in the signature s_{ic} through the MAC mechanism. When more than 2ν nodes have passed the consistency verification, a consensus is reached. After reaching a consensus, CA_r reads any t partial signature s_{ic} from the commit identifier to the second partial signature signaling $S = \sum s_{ic}$. The whole weighted identifier is finally determined as (G, r, S).

(7) CA_r calculates the hash value H_r with weight identification Φ and save H_r in the block. AS sends weight identification Φ to identification requester R through the secret channel.

4 Experimental Discussions and Analysis

4.1 Experimental Environment and Parameters

To verify the effectiveness of the proposed method, a series of experiments are conducted in this section. The hardware configurations are Intel (R) core (TM) i3-3110M 2.40 GHz CPU with 16 GB memory host. In addition, the software environment is Windows 10 while the Geth client is downloaded for the implementation of blockchain. All the relevant algorithms are programmed with Microsoft VC + + 6.0. In the experiment, e_1, e_2 and d are all set to the integer of more than 120 digits while the number of members n is 60. The microgrid test system based on blockchain is shown in Fig. 4.

All the power data from electric vehicles, photovoltaic power plants and loads and so on are collected by the smart meters and intelligent control equipment. Then, the collected data are uploaded to the data acquisition base station. As the authentication node in the blockchain, the base station can communicate with other authentication nodes. The base station is mainly responsible for issuing weight identification to the smart meter information added to the region for identity registration and authentication, and is also responsible for implementing access control policies. It is assumed that the microgrid test system includes 6 base stations, 20 electric vehicles, one photovoltaic power plant, one hydropower station and one wind power plant.

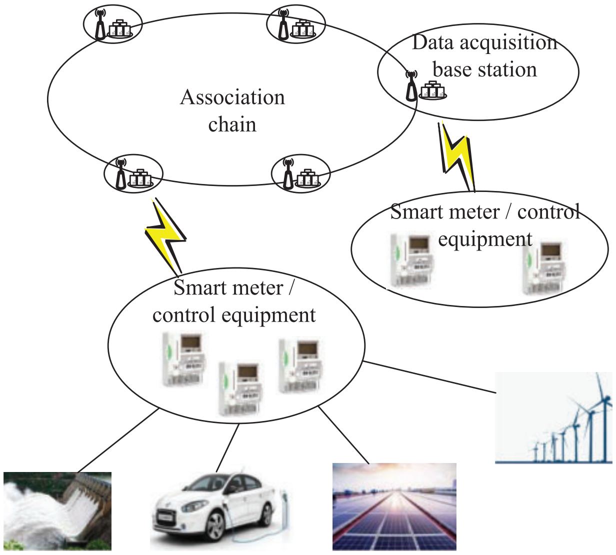
Association chain
Data acquisition base station
Smart meter / control equipment
Smart meter / control equipment

Figure 4: Microgrid test system with blockchain

4.2 Time Consumption of Private Key Update

In the key distribution stage, the setting of threshold value has a great impact on the time consumption in the process of private key update. To see the performance of the proposed algorithm in private key updating, we compare our method with [17] which adopts the traditional chained topology. The corresponding results are shown in Fig. 5, where the threshold values are 10, 20, 30, 40, 50, 60 and 70, respectively.

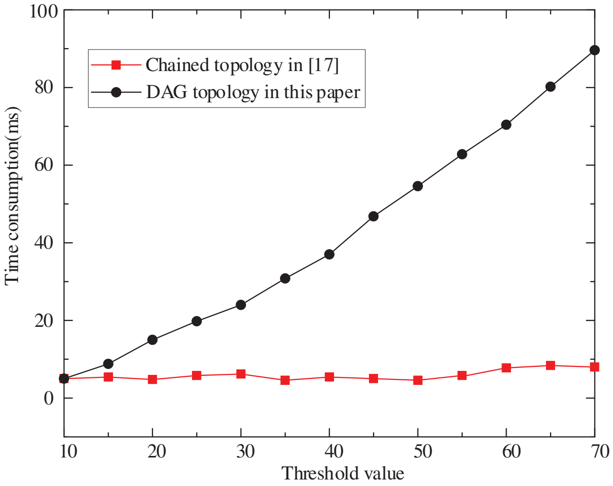

Figure 5: Comparison of private key update process in time consumption

According to Fig. 5, with the threshold value increases, our method can retain a stable value within 5 min which is with fewer time consumption compared with the traditional chained topology. In addition, it can also be seen from Fig. 5 that there is an insignificant impact of the threshold value on the time consumption with our proposed method. The reason is that our proposed method adopts a weighted key distribution scheme based on the secret sharing idea. Then the distribution of keys with different weights to the CA institutions with different authorities can be realized. Besides, the adoption of the node trust mechanism can meet the regular private key update demands of the members in CA. In [17], the main calculation step in private key update is to calculate a polynomial with a high degree. It means that there exists a high time complexity in the method proposed in [17]. Conversely, the main calculation step of the proposed method is $\phi_{kc}^{(T)} = A_k^{(T)} e_2 \bmod u_c$, where $A_k^{(T)}$ is a randomly selected integer, and only one multiplication is required in this step. Hence, with our proposed method, the corresponding time complexity is lower compared with [17]. In other words, compared with the chained topology in [17], the time consumption of private key update with our proposed method is shorter, which means that the private key issuance efficiency considering the trusted node is higher.

4.3 Time Consumption of Private Key Update

To demonstrate the blockchain construction speed with the proposed method, the methods proposed in [14,17] are adopted as the comparisons in this subsection. The traditional chained topology is adopted in both two comparative methods. It should be pointed out that unlike [17,14] considers the node trust in the blockchain construction process. Fig. 6 shows the results of chain construction time consumption with the number of blockchain nodes increasing from 1 to 25.

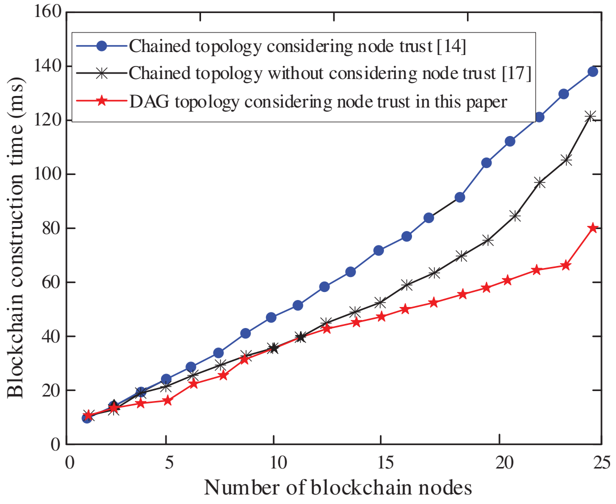

Figure 6: The relationship between chain construction time and the number of blockchain nodes

As can be seen from Fig. 6, the chain construction time increases for all methods as the number of nodes in the blockchain increases. However, our proposed method has a shorter time consumption compared to the traditional methods in the chain construction process. In addition, when the number of nodes increases from 1 to 25, the link construction time using our proposed method only increases by 70 milliseconds. Our proposed method achieves a 46.15% and 36.36% reduction in chain construction time compared with [14,17], respectively. This is because the proposed method adopts the directed acyclic graph consensus mechanism which only uses partial nodes rather than all nodes to authenticate the data storage operation of a single node, enabling concurrent data storage operations and improving corresponding data storage efficiency.

4.4 Data Throughput

The data throughput test process is conducted in the Ganache development environment. The number of test accounts is 15 by default. In this subsection, the concurrent virtual user threshold and the maximum number of concurrent transactions are set as 15. Different operations are randomly performed, such as deploying smart contracts, transaction chaining and contract invocation. We analyzed the transaction execution time during system operation within 120 s and the results are presented in Fig. 7.

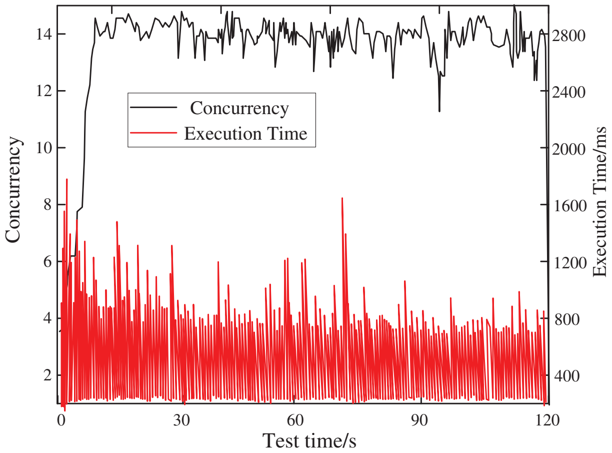

Figure 7: Virtual user concurrency test results

Fig. 7 shows that during the 120-s operation, the average number of concurrent virtual users is 14.19, with an average transaction execution time of 823.26 milliseconds. In addition, Fig. 7 also shows that the maximum transaction execution time is 1680 ms, and the minimum transaction execution time is 753 milliseconds. It can be found the number of request per second (RPS) of system setting is slightly less than the setting value of concurrency threshold. The reason is that processing time in consensus mechanism is increased due to the variable assignment and parameter return operations. The test results of the virtual user concurrency indicate that the number of concurrent users has a minimal impact on the blockchain system, but the fluctuations in transaction execution time suggest that the stochastic feature of network delay may have a greater impact on the transaction execution time of the blockchain system.

To further demonstrate the data storage efficiency of the proposed method, we compared the data throughput of our proposed method with methods proposed in [14,17]. Reference [14] proposed an efficient lightweight integrated blockchain model, which integrates resources into a public blockchain by generating an overlay network. However, since the blockchain requires consensus authentication from most nodes, the data storage is inefficient and the data throughput is low. Reference [17] proposed a blockchain-based secure power transaction mechanism for the wireless network in the smart grid, which still requires the consensus authentication of most nodes. Thus, as the number of blockchain nodes increases, the data throughput in [17] declines slightly, and blockchain authentication takes up a significant amount of data storage time.

As shown in Fig. 8, with the increase in the number of blockchain nodes, the system data throughput shows an upward trend. However, the proposed method exhibits higher data throughput than the other methods. When the number of blockchain nodes reaches 25, the throughput of the proposed method is close to 300 kbps. Given the same number of block nodes, the proposed method achieves the improvements of 33.51% and 8.94%, respectively, in average data throughput compared with the consensus methods in [14] and [17]. This can be attributed to the DAG-based blockchain consensus algorithm adopted in this paper, which has higher data storage efficiency.

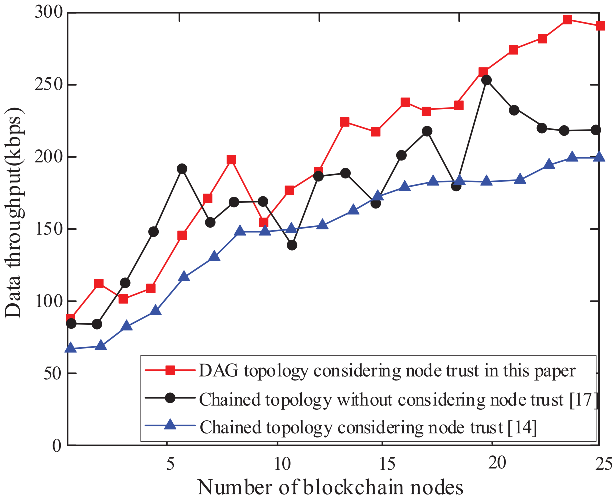

Figure 8: Comparison of different methods in data throughput

4.5 Data Security Test

To verify the security of the proposed method, different tampering methods are tested in the Ganache development environment, and their success rates are evaluated. Fig. 9 illustrates the tampering success rate of microgrid data with the increase of blockchain nodes. Table 1 demonstrates the effectiveness of data storage security with different consensus mechanisms when the storage nodes is 25.

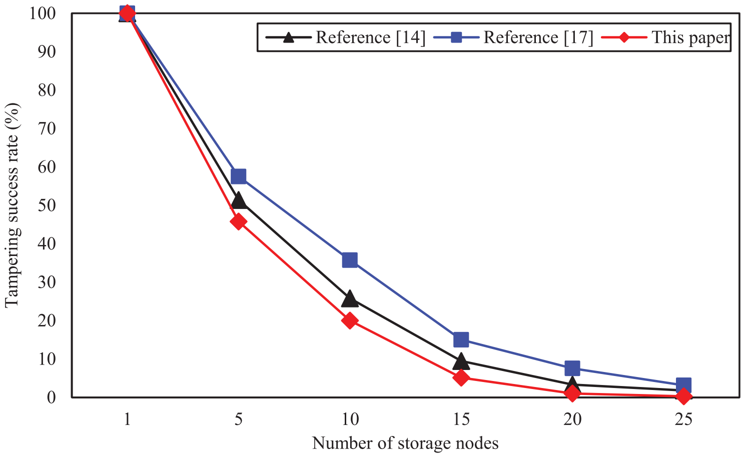

Figure 9: Tampering success rate with increased storage nodes

Table 1: Data security values in different methods

Methods	Chained topology considering node trust [14]	Chained topology without considering node trust [17]	DAG topology considering node trust in this paper
Data security (%)	98.21	96.87	99.73

From Fig. 9, it is evident that as the number of storage nodes increases, all the tampering success rates with different consensus mechanisms will be rapidly decreased. However, since our proposed method fully takes the trust of storage nodes into account, the proposed method has the lowest tampering success rate when the same number of storage nodes are considered. In addition, because that the proposed method adopts the DAG topology, the fastest decline speed of tampering success rate can be guaranteed compared to other two methods. Moreover, as indicated in Table 1, although the data security exceeds 95% for all three methods, but the method proposed by [17] achieves only 96.87% because the node trust is not considered in blockchain construction process. Among all the methods, our proposed method achieves the highest data security. Specifically, our proposed method achieves a gain of 1.5% and 2.9%, respectively, over the other two methods in data security. The experimental results show that the proposed method can effectively enhance the resistance ability against cyber attacks.

5 Conclusions

The efficient and secure storage of data is crucial for many data-driven applications in microgrids. Based on the blockchain technology, this paper proposes a secure storage management method for microgrid data based on node trust and directed acyclic graph (DAG) consensus mechanism in Power Internet of Things. The proposed DAG consensus algorithm addresses the time-consuming issue in traditional chain topology consensus mechanisms, improving storage efficiency. Considering the differences in the resistance ability of storage nodes against cyber attacks, the trusts of candidate nodes are introduced into the hash value update of the blockchain header. The nodes with high trusts are then given higher priority data storage rights. The experimental analysis of the built microgrid data storage platform shows the private key update time of the proposed method does not exceed 5 milliseconds, and the number of concurrent users has limited effects on the system stability. The number of blockchain nodes does not exceed 25, the blockchain construction time does not exceed 80 ms, and the data throughput is close to 300 kbps, which suggests our proposed method has outperformed other methods. It is worth noting that our proposed method does not match the corresponding smart contracts, the core part of blockchain, and its data security storage performance also has room for improvement. For this reason, matching smart contracts will be developed under the blockchain architecture, and microgrid data storage methods will be continuously optimized to improve the storage quality of grid data in future research.

Acknowledgement: All authors contributed to the study conception and research design. Material preparation, data collection and analysis were performed by Jian Shang, Runmin Guan and Wei Wang. The first draft of the manuscript was authored by Jian Shang and all authors commented on previous versions of the manuscript. The final manuscript was read and approved by all authors.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Tang and K. Shi, "Data privacy protection technology of wearable-devices," *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 2, pp. 2973–2980, 2021.

[2] M. Yang, J. Guo, Z. Zhao, T. Xu and L. Bai, "Teenager health oriented data security and privacy protection research for smart wearable device," *Procedia Computer Science*, vol. 174, no. 7, pp. 333–339, 2020.

[3] W. Zhou, Y. Wang and J. Wen, "A trust-based evaluation model for data privacy protection in cloud computing," *International Journal of High Performance Computing Applications*, vol. 14, no. 2, pp. 147–156, 2019.

[4] Y. Duan, Z. Lu, Z. Zhou, X. Sun and J. Wu, "Data privacy protection for edge computing of smart city in a DIKW architecture," *Engineering Applications of Artificial Intelligence*, vol. 81, no. 5, pp. 323–335, 2019.

[5] Y. Zhang, C. Peng, S. Xie and X. Du, "Deterministic network calculus-based H∞ load frequency control of multiarea power systems under malicious DoS attacks," *IEEE Transactions on Smart Grid*, vol. 13, no. 2, pp. 1542–1554, 2022.

[6] H. R. Bokkisam, S. Singh, R. M. Acharya and M. P. Selvan, "Blockchain-based peer-to-peer transactive energy system for community microgrid with demand response management," *CSEE Journal of Power and Energy Systems*, vol. 8, no. 1, pp. 198–211, 2022.

[7] B. Wu, X. Chen, C. Zhang, Z. Mei and T. Yan, "Privacy-protection path finding supporting the ranked order on encrypted graph in big data environment," *IEEE Access*, vol. 8, no. 5, pp. 214596–214604, 2020.

[8] Z. Ning, S. Sun, X. Wang, L. Guo, G. Wang *et al.,* "Intelligent resource allocation in mobile blockchain for privacy and security transactions: A deep reinforcement learning based approach," *Science China-Information Sciences*, vol. 64, no. 6, pp. 1–16, 2021.

[9] H. Yi, "Securing instant messaging based on blockchain with machine learning," *Safety Science*, vol. 120, no. 1, pp. 6–13, 2019.

[10] D. Mashima, A. Serikova, Y. Cheng and B. Chen, "Towards quantitative evaluation of privacy protection schemes for electricity usage data sharing," *ICT Express*, vol. 4, no. 1, pp. 35–41, 2018.

[11] M. Bouchaala, C. Ghazel and L. A. Saidane, "Enhancing security and efficiency in cloud computing authentication and key agreement scheme based on smart card," *Journal of Supercomputing*, vol. 112, no. 1, pp. 1–26, 2021.

[12] T. Kanwal, A. Anjum, S. Malik, H. Sajjad, A. Khan *et al.,* "A robust privacy preserving approach for electronic health records using multiple dataset with multiple sensitive attributes," *Computers & Security*, vol. 105, no. 1, pp. 1–21, 2021.

[13] S. Seybou, F. Essaf and M. Mbyamm, "Privacy protection issues in blockchain technology," *International Journal of Information Security*, vol. 17, no. 2, pp. 124–131, 2019.

[14] Y. Ren, Y. Leng, F. Zhu, J. Wang and H. Kim, "Data storage mechanism based on blockchain with privacy protection in wireless body area network," *Sensors*, vol. 19, no. 10, pp. 2395–2402, 2019.

[15] Y. Chen, H. Xie, K. Lv, S. Wei and C. Hu, "DEPLEST: A blockchain-based privacy-preserving distributed database toward user behaviors in social networks," *Information Sciences*, vol. 501, no. 2, pp. 100–117, 2019.

[16] W. Lu, Z. Ren, J. Xu and S. Chen, "Edge blockchain assisted lightweight privacy-preserving data aggregation for smart grid," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1246–1259, 2021.

[17] Z. Liu, D. Wang, J. Wang, X. Wang and H. Li, "A blockchain-enabled secure power trading mechanism for smart grid employing wireless networks," *IEEE Access*, vol. 8, no. 3, pp. 177745–177756, 2020.

[18] W. Dong, Q. Yang, W. Li and A. Y. Zomaya, "Machine-learning-based real-time economic dispatch in islanding microgrids in a cloud-edge computing environment," *IEEE Internet of Things Journal*, vol. 8, no. 17, pp. 13703–13711, 2021.

[19] X. Li, Y. Mei, J. Gong and F. Xiang, "A blockchain privacy protection scheme based on ring signature," *IEEE Access*, vol. 8, no. 7, pp. 76765–76772, 2020.

[20] H. Wang, C. Wang, Z. Shen and K. Liu, "A MADM location privacy protection method based on blockchain," *IEEE Access*, vol. 9, no. 7, pp. 27802–27812, 2021.

[21] B. Luo, X. Li, J. Weng, J. Guo and J. Ma, "Blockchain enabled trust-based location privacy protection scheme in VANET," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2034–2048, 2020.

[22] J. Cheng, J. Li, N. Xiong, M. Chen, H. Guo *et al.,* "Lightweight mobile clients privacy protection using trusted execution environments for blockchain," *Computers, Materials & Continua*, vol. 65, no. 3, pp. 2247–2262, 2020.

[23] M. Ghorbanian, S. H. Dolatabadi, P. Siano, I. Kouveliotis-Lysikatos and N. Hatziargyriou, "Methods for flexible management of blockchain-based cryptocurrencies in electricity markets and smart grids," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4227–4235, 2020.

[24] S. Zou, J. Xi, H. Wang and G. Xu, "CrowdBLPS: A blockchain-based location-privacy-preserving mobile crowdsensing system," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4206–4218, 2020.

[25] Y. Wen, J. Liu, W. Dou, X. Xu, B. Cao *et al.,* "Scheduling workflows with privacy protection constraints for big data applications on cloud," *Future Generation Computer Systems*, vol. 108, no. 2, pp. 1084–1091, 2020.

[26] J. Li, Z. Zhou, J. Wu, J. Li, S. Mumtaz *et al.,* "Decentralized on-demand energy supply for blockchain in Internet of Things: A microgrids approach," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1395–1406, 2019.

[27] Y. Wang, X. Liang, X. Hei, W. Ji and L. Zhu, "Deep learning data privacy protection based on homomorphic encryption in AIoT," *Mobile Information Systems*, vol. 6, no. 2, pp. 1–11, 2021.